Monte Carlo Status and (mostly) Questions....

Mark Messier

IU Software Workshop

11 NOV 2005

Reminder of the Monte Carlo "design"

- GEANT3 based simulation using dd_geant/rcp from D0
- Offline C++ packages linked to backend: output is ROOT format data which includes the truth hits and particle stacks.
- List of offline packages includes: CkovDigitizer, DCDigitizer, E907MCInterface, E907MCInterface, Geometry, MCClasses, MCTruth, MWPCDigitizer, RICHDigitizer,TOFDigitizer

Status

- Fallen somewhat into disrepair
- Raja is only person to run it in recent times and only linked to the FORTRAN side. C++ side not exercised in quite sometime
- Still, quite a lot is working
  - Much of digitization still working (see later in talk for details)
  - Geometry includes survey (Raja)
  - Output to root format still works

What do we need to do to make the MC a useful tool for testing reconstruction?

I. Make it easier to run

- MC requires a great deal of configuration. Difficult for non-expert to understand and manage
- I recently added a script "e907mc.sh" which allows MC to be run with a single command. Takes care of all environment variables and soft links
- Setup uses SRT and should be site-independent

```
e907mc.sh usage:
  -d,--debug           : Run in debug mode
  -s,--stdhep [file]   : Input kinematics in stdhep format
  -r,--rzout [file]    : Output GEANT zebra file
  -g,--histo [file]    : Output PAW histogram file
  -p,--prompt          : Run in interactive mode
  -t,--target [tgt]    : Select target. Options are:
                           wheel = target wheel
                           cryo  = cryo target
                           numi  = NuMI target
  -v,--verbose         : Increase debugging text output
  -h,--help            : Print this message
```

Please use it to keep it working!

II. Confirm that is still does at least what it used to do

I'm having some troubles getting output (hits/digits) into root file
May be something simple I'm doing wrong at or something simple wrong with the code. But we need to get the root output at least to the state it was in > 1 year ago to make make progress.

Digitization status:

- BC's (new!)
- BCKOV (none?)
- TPC (overly simple)
- Chambers (exists, overly simple)
- CKOV (started)
- TOF (simple model, may require some tuning)
- RICH (works, but much too clean)
- ECAL (none?)
- HCAL (none?)

III. Simple level of tuning of digitization algorithms

Get all digitizers to state where output TDC's, ADC's, etc are in same range as data. Will allow reconstruction of MC events

Add TPC distortions using algorithms used to take distortions out run in reverse

Get DC efficiencies in roughly correct range

This should make data "reconstructable?"

IV. Design question: How to handle RunInfo data?

Currently we rely on database for a large amount of information about run conditions:

- Target choice
- Beam momentum
- JGG & ROSY Fields
- Drift velocity and offset
- Noisy/dead channels

If same algorithms are to work on data and MC, we need this level of information to be served to MC runs. How?

- “RunInfo object” in data stream which contains all this information?
- Reference “real” run number?
- Other options?

V. Design question: How to handle beam simulation?

Like to preserve use of 3rd party packages (FLUKA, RQMD, ...) for interaction **but** would like to have digitization/simulation of detector response to beam particle

Solutions:

[1] Forget get it and just use GFLUKA/GCALOR for target simulations
Pro: Simple! Easy to maintain
Con: GFLUKA/GCALOR are poor physics models

[2] Use stdhep file to list interactions. Stop tracking beam particle at target face. Translate/rotate kinematics from stdhep file to location of beam on target.

[3] Others??